



高等学校本科规划教材

微机原理与嵌入式系统基础

赵全良 马 博 孟李林 编著



西安电子科技大学出版社
<http://www.xduph.com>

面向 21 世纪高等学校本科规划教材

微机原理与嵌入式系统基础

赵全良 马 博 孟李林 编著

西安电子科技大学出版社

2010



内 容 简 介

本书以 ARM7 TDMI 处理器为模型机讲述了微型计算机原理,同时讲述了常用的嵌入式微控制器 LPC2132 的基本组成结构和工作原理。

本书共分为 8 章,主要包括:计算机与嵌入式系统基础知识、ARM7 体系结构、ARM7 指令系统与汇编语言程序设计、存储器原理与扩展、输入/输出与中断技术、LPC2132 系统结构与资源原理以及 ARM 开发工具 ADS。

本书结构合理,概念清晰,讲解深入浅出,既可作为高等院校电类专业“微型计算机原理”和“嵌入式设计基础”等课程的教材,也可作为广大嵌入式系统设计爱好者的入门学习教材。

★本书配有电子教案,需要者可登录出版社网站,免费下载。

图书在版编目(CIP)数据

微机原理与嵌入式系统基础/赵全良,马博,孟李林编著. —西安:西安电子科技大学出版社,2010.2
面向 21 世纪高等学校本科规划教材

ISBN 978-7-5606-2396-2

I. 微… II. ①赵… ②马… ③孟…

III. ①微型计算机—高等学校—教材 ②微型计算机—系统设计—高等学校—教材 IV. TP36

中国版本图书馆 CIP 数据核字(2010)第 013208 号

策 划 云立实

责任编辑 雷鸿俊 云立实

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2010 年 2 月第 1 版 2010 年 2 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 16.375

字 数 385 千字

印 数 1~3000 册

定 价 23.00 元

ISBN 978 - 7 - 5606 - 2396 - 2/TP · 1202

XDUP 2688001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜,谨防盗版。

前 言

嵌入式系统应用已广泛地渗透于现实社会的科研、生产、军事和日常民用中，并且还在向更广的范围和深度快速发展。嵌入式系统的研发涉及到微机原理、嵌入式系统设计和软件设计等相关的基础知识，需要设计人员有较高的综合能力。

以往的教材多以 Intel 公司的 X86 为模型机讲解微机原理，着重讲解微型计算机的基本概念和原理；而在学习嵌入式系统设计时，则认为读者已具备了微机原理知识，仅补充学习嵌入式微处理器原理。在现有的课程体系中，需要通过不同的课程，来学习微机原理和嵌入式系统设计。

本书是作者基于多年的微机原理和嵌入式系统设计教学与科研的积累而写成的，结合了微机原理基本概念和嵌入式微控制器原理与应用，力图使读者较快地、有针对性地学习微机原理与嵌入式系统设计，缩短读者学习的周期，旨在为学习嵌入式系统设计的人员提供入门级的微机原理与嵌入式系统设计基础知识。

本书以最常用的嵌入式微处理器 ARM7 为模型机，系统地讲解了微型计算机的基本概念、组织结构和工作原理，进而讲解了广泛使用的恩智浦公司的嵌入式微控制器 LPC2132 的基本原理，包括 LPC2000 系列微控制器的指令系统、常用资源的原理和应用等。基础较好的读者在学习过程中，可以结合恩智浦公司给出的 LPC2132 数据手册深入地学习嵌入式系统设计。

学习本课程时要注意基本概念和基本理念的建立，更要注意在实验中加强概念的理解。本书使用周立功公司的 EasyARM2131 开发板作为学习实验平台，该平台价格低廉，其低层次的系统封装更有助于对嵌入式系统工作原理的理解。学习嵌入式系统设计至少要掌握一种开发工具，本书给出了 ARM 公司的嵌入式开发工具 ADS 的简介，便于读者在实验过程中初步掌握该工具的使用。

本书写作时，作者致力于内容简明扼要，概念准确、清晰。为使教材内容精练，缩减了实验性的内容。但须知，本课程的实践性很强，只有在实践中不断总结、领会，才能加强巩固基础理论知识，建立微型计算机原理概念和嵌入式系统设计理念，才能在嵌入式系统设计的道路上走得更远。

本书第 1、2、6 章和第 7 章由赵全良编写，第 3、4 章和第 8 章由马博编写，第 5 章由孟李林编写。赵全良担任主编并负责本书的大纲拟定和全书的统稿工作。

在本书的编写过程中，得到了周立功公司和周立功先生的大力支持，在此谨向周立功先生致以诚挚的感谢。

本书的编写和出版得到了西安邮电学院计算机系领导的热切关心，并得到有关教师多方面的帮助与鼓励，在此表示感谢。

最后更要感谢我们的家人，正是他们的理解与支持才使我们可以集中精力致力于计算机与嵌入式系统的教学与科研事业之中。

限于作者的水平，书中疏漏与不当之处在所难免，恳请读者批评指正。

作者 E-mail: zqlcba@163.com(赵全良); mabo124@sohu.com(马博)

编 者

2009 年 12 月

目 录

第1章 计算机与嵌入式系统基础知识	1	2.3.3 Thumb 状态下的寄存器	45
1.1 计算机基础知识	1	2.4 ARM7 当前程序状态寄存器(CPSR)	46
1.1.1 引言	1	2.4.1 CPSR 中的控制位	47
1.1.2 计算机的发展	1	2.4.2 CPSR 中的状态标志位	48
1.1.3 冯·诺伊曼计算机体系结构	2	2.5 ARM7 的异常与异常处理	49
1.1.4 计算机系统	5	2.5.1 ARM7 的异常	49
1.1.5 计算机的工作过程	5	2.5.2 ARM7 的异常处理	49
1.2 数制和计算机中的数值表示与处理	6	2.6 ARM7 的存储系统	52
1.2.1 无符号数的十进制、十六进制、 二进制表示法与运算	7	2.6.1 ARM7 处理器寻址空间	53
1.2.2 带符号数的原码、反码和补码表示 ..	11	2.6.2 存储器中数据组织的形式	53
1.2.3 补码的运算、溢出及其判断方法	14	本章小结	54
1.2.4 真值与机器数	18	习题	55
1.3 计算机中的其他信息编码	19	第3章 ARM7 TDMI 指令系统	57
1.3.1 BCD 编码	19	3.1 ARM7 TDMI 编程模型	57
1.3.2 字符 ASCII 编码	21	3.2 ARM7 TDMI 的寻址方式	57
1.4 嵌入式系统概述	23	3.2.1 数据处理指令操作数寻址方式	58
1.4.1 嵌入式系统的基本概念	23	3.2.2 存储器访问指令操作数寻址方式	59
1.4.2 嵌入式处理器的分类与发展概况	24	3.3 ARM7 TDMI 指令的条件执行	61
1.4.3 常见的几种嵌入式操作系统介绍	26	3.4 ARM 指令集	63
本章小结	29	3.4.1 ARM 指令的基本格式	63
习题	30	3.4.2 ARM 存储器访问指令	65
第2章 ARM7 体系结构	32	3.4.3 ARM 数据处理指令	72
2.1 概述	32	3.4.4 ARM 分支指令	77
2.1.1 RSIC 处理器与 CISC 处理器结构 特征简介	33	3.4.5 ARM 杂项指令	78
2.1.2 ARM7 TDMI	33	3.4.6 ARM 软中断指令	81
2.1.3 ARM7 处理器的三级流水线	35	3.4.7 ARM 伪指令	81
2.2 ARM7 的状态与模式	37	3.5 Thumb 指令集	84
2.3 ARM7 的寄存器	39	3.5.1 Thumb 存储器访问指令	85
2.3.1 ARM 状态下的寄存器	39	3.5.2 Thumb 数据处理指令	86
2.3.2 ARM 处理器特别功能寄存器	41	3.5.3 Thumb 分支指令	87
		3.5.4 Thumb 软中断指令	87
		3.5.5 Thumb 伪指令	87

本章小结	88	5.5.2 地址空间划分及存储器连接.....	134
习题	89	5.6 存储器扩展.....	139
第4章 ARM7 TDMI 汇编语言		5.6.1 位扩展.....	139
程序设计	91	5.6.2 字扩展.....	140
4.1 汇编语言的基本概念	91	5.6.3 字位同时扩展.....	140
4.2 ARM 汇编程序的组成与结构	92	本章小结.....	142
4.2.1 汇编器伪指令	94	习题.....	143
4.2.2 ARM 汇编程序中的数据定义	94	第6章 输入/输出与中断技术	144
4.3 汇编语言程序设计的基本方法	97	6.1 计算机接口概述.....	144
4.3.1 结构化程序设计的基本概念	97	6.2 输入/输出接口电路.....	145
4.3.2 汇编语言程序设计的流程	98	6.2.1 输入/输出接口电路的基本功能与 端口分类.....	145
4.3.3 顺序程序设计	99	6.2.2 I/O 端口的编址	146
4.3.4 分支程序设计	100	6.3 CPU 与外设的数据传输控制方式.....	146
4.3.5 循环程序设计	101	6.3.1 无条件数据访问方式.....	146
4.3.6 子程序设计	105	6.3.2 状态查询数据访问方式.....	147
4.4 汇编语言和 C 语言交叉编程.....	109	6.3.3 中断数据访问方式.....	149
4.4.1 汇编程序与 C 程序间变量互访.....	109	6.3.4 DMA 方式数据访问传输	149
4.4.2 汇编程序调用 C 程序.....	110	6.4 ARM 中的 GPIO	151
4.4.3 C 程序调用汇编程序.....	111	6.4.1 概述.....	151
4.4.4 C 程序中内嵌汇编代码.....	112	6.4.2 GPIO 原理及端口寄存器说明	152
本章小结	113	6.4.3 GPIO 应用举例	154
习题	114	6.5 中断技术.....	155
第5章 存储器原理与扩展	115	6.5.1 基本概念.....	155
5.1 概述	115	6.5.2 中断系统的功能.....	156
5.1.1 半导体存储器的分类	116	6.5.3 中断处理过程.....	158
5.1.2 半导体存储器的常用术语	117	6.6 ARM 中断系统基础.....	160
5.1.3 半导体存储器的主要性能指标	118	6.6.1 ARM 中的中断源.....	160
5.2 随机存储器	119	6.6.2 ARM 中的向量中断控制器.....	161
5.2.1 静态 RAM	119	6.6.3 ARM 的中断响应过程.....	162
5.2.2 动态 RAM	124	6.6.4 ARM 的中断返回.....	162
5.3 只读存储器	127	6.6.5 ARM 的外中断.....	163
5.3.1 掩膜式 ROM	127	本章小结.....	170
5.3.2 一次编程式 ROM	128	习题.....	170
5.3.3 多次编程式 ROM	128	第7章 LPC2132 系统结构与资源	
5.4 Flash 存储器.....	130	原理	171
5.4.1 类型及特点	130	7.1 LPC2132 芯片简介	171
5.4.2 芯片介绍	132	7.1.1 特性.....	171
5.5 存储器与 CPU 连接.....	133	7.1.2 器件信息.....	172
5.5.1 连接时应注意的问题	133		

7.1.3 结构原理	172	7.6.2 ARM 的中断源	218
7.1.4 片内存储器	173	7.6.3 VIC 结构原理	219
7.1.5 LPC2132 管脚	174	7.6.4 端口寄存器与操作	220
7.2 LPC2132 地址空间分配	179	7.6.5 中断方式应用举例	228
7.2.1 存储器映射	180	本章小结	233
7.2.2 存储器重映射	180	习题	234
7.2.3 引导模块	181	第 8 章 ARM 开发工具 ADS	236
7.3 管脚连接模块	183	8.1 ADS 简介	236
7.3.1 概述	183	8.1.1 ADS 环境下 ARM 程序开发方式	236
7.3.2 结构原理	184	8.1.2 ADS 工具介绍	238
7.3.3 端口寄存器与操作	184	8.2 ADS 中的工程管理工具	
7.4 定时器/计数器	186	CodeWarrior IDE	239
7.4.1 概述	186	8.2.1 ADS 系统中的文件类型	239
7.4.2 结构原理	187	8.2.2 工程模板	239
7.4.3 管脚描述	189	8.2.3 创建工程	240
7.4.4 端口寄存器	189	8.2.4 工程管理	243
7.4.5 定时器/计数器应用举例	196	8.3 ADS 中的调试工具 AXD	243
7.5 通用异步收发器	198	8.3.1 调试工具条	244
7.5.1 概述	198	8.3.2 AXD 调试器中常用的调试窗口	245
7.5.2 异步通信及其协议	199	8.4 ADS 环境汇编程序设计实例	247
7.5.3 LPC2000 器件 UART 特性	202	8.4.1 编辑实例	247
7.5.4 结构原理	202	8.4.2 工程的调试	249
7.5.5 管脚描述	206	8.4.3 配置 AXD IDE 调试环境	250
7.5.6 端口寄存器与操作	207	本章小结	251
7.5.7 串行通信应用举例	214	习题	252
7.6 向量中断控制器	217	参考文献	253
7.6.1 概述	217		

第1章 计算机与嵌入式系统基础知识

冯·诺依曼体系结构计算机仍然是现代电子计算机的主流。通过本章的学习,应了解计算机的发展历史,掌握冯·诺依曼计算机体系结构的要点,理解计算机的系统构成和工作原理,掌握基本数据和信息在计算机中的存储形式和处理方式,建立嵌入式系统的基本概念。

1.1 计算机基础知识

电子计算机是以微电子学、数学、系统论为基础的现代高科技综合产物,它已广泛地深入到我们科研、生产和生活的各个角落。本章简要地介绍计算机的系统组成、工作原理、信息在计算机中的表示方法和处理方式。

1.1.1 引言

电子计算机于1946年在美国宾西法尼亚大学诞生,初期主要用于科学计算,随着科学技术的发展,电子计算机更广泛地应用于过程控制和信息处理等领域。在我们的日常工作和生活中,无处不见它的身影。我们见到的计算机有的以独立的方式存在,如PC机、工作站、小型机等;更多的则是以设备嵌入的方式被应用,如MP4、手机、网络设备、智能家电,甚至导弹、飞行器控制等。美国通用公司早在几年前就宣布,其每年以汽车电子方式销售的计算机的运算能力远远大于以计算机为主业的IBM公司。

嵌入式系统应用是计算机科技发展的重要分支。其天地之广阔,市场之巨大是通用型计算机无法比拟的,所需的专业性人才在可预见的近十年内将供不应求。

1.1.2 计算机的发展

到目前为止,计算机按实现工艺的发展已经历了四代。

1) 第一代——电子管计算机

第一代电子计算机以电子管为逻辑元件,体积大、功耗高、造价高、运算速度慢,编程语言以机器语言和汇编语言为特征,使用不方便,主要应用于科学运算。

在计算机发展的早期阶段,人们认为计算机的主要用途是快速计算,软件编程简单,不存在系统化的设计方法,开发过程没有任何管理,程序的质量完全依赖于程序员个人的技巧。

2) 第二代——晶体管计算机

第二代计算机问世于1954年,由晶体管取代了电子管。与电子管相比,晶体管具有体积小、重量轻、寿命长、效率高、功耗低等特点,并把计算速度从几千次每秒提高到几十

万次每秒, 编程语言已使用高级语言。

第二代计算机还是主要用于科学与工程计算, 使用 FORTRAN 语言、汇编语言编写程序。在后期出现了操作系统的雏形, 对计算机系统资源进行管理, 为计算机使用者提供支持。

3) 第三代——集成电路计算机

第三代计算机诞生于 1964 年, 由集成电路取代了晶体管。与晶体管相比, 集成电路的体积更小, 功耗更低, 可靠性更高。第三代计算机由于采用了集成电路, 体积因此大大缩小, 计算速度从几十万次每秒提高到上千万次每秒, 价格也不断下降, 并广泛地使用高级语言编程。

计算机软件发展的第二阶段跨越了从 20 世纪 60 年代中期到 70 年代末期的十余年, 多用户系统引入了人机交互的新概念, 实时系统能够从多个来源收集、分析和转换数据, 从而使得进程的控制和输出的产生以毫秒而不是分钟来进行, 在线存储的发展产生了第一代数据库管理系统。

这个时期, 出现了软件产品和“软件作坊”的概念, 设计人员开发程序不再像早期那样只为自己的研究工作需要, 而是为了用户更好地使用计算机。人们开始采用“软件工程”的方法来解决“软件危机”问题, 操作系统作为系统资源的管理者成为了热门, 其代表就是著名的 UNIX 操作系统。

4) 第四代——大规模、超大规模集成电路计算机

第四代计算机是采用大规模集成电路制造的计算机, 高度的集成化使得计算机的中央处理器和其他主要功能部件可以集中到同一块集成电路中, 这就是人们常说的“微处理器”。第一台微处理器“4004 芯片”于 1971 年由英特尔公司研制成功, 这块集成了 2300 个晶体管的芯片的面积只有 $4.2 \times 3.2 \text{ mm}^2$, 功耗处于毫瓦级; 如果相同功能的处理器还用 20 世纪 50 年代初的电子管来做的话, 电路板需要数十平方米, 功耗也会以千瓦来计。此后, 微处理器的发展如同乘上了高速列车, 大约每隔 18 个月, 性能价格比就翻一番。

随着计算机技术的不断更新与发展, 微型计算机神奇地闯入了人们的生活, 现在以低廉的价格就可以获得强大的计算能力。当价格不再是阻拦计算机普及的门槛时, 降低计算机的易用性就显得十分重要。由于 UNIX 操作系统本身的特点, 使得其不太适合于运行在个人计算机上, 人们迫切需要一种新的平民化的操作系统。微软公司适时地进入了这一领域, 将购买来的 CP/M 演变成为 MS-DOS, 以至后来的视窗操作系统 Windows, 并凭借 Windows 成为个人计算机操作系统领域的主流。应用开发编程环境更为友好, 出现了许多面向不同应用的计算机语言系统。目前, 计算机的软硬件, 尤其是微型计算机还在迅猛发展之中。

1.1.3 冯·诺伊曼计算机体系结构

冯·诺伊曼(John Von Neumann, 1903—1957), 著名数学家, 1903 年 12 月 28 日生于匈牙利布达佩斯, 1957 年 2 月 8 日在华盛顿因癌症去世。冯·诺伊曼的父亲马克斯是位富有的犹太银行家, 母亲也受过良好的教育。冯·诺伊曼 1921 年进入柏林大学, 1923 年进入瑞士苏黎世联邦工业大学学习化学, 1925 年取得化学工程师的资格。1926 年春在布达佩斯大学获博士学位。1927 年任柏林大学讲师。1929 年任汉堡大学讲师。



1930年应聘到普林斯顿大学任教,1933年成为新建的普林斯顿高等研究所教授。在第二次世界大战期间移居美国,曾任研制原子弹的顾问,并参加研制计算机。1954年成为美国原子能委员会委员并移居华盛顿。

冯·诺伊曼是20世纪最著名的数学家之一,在数学理论和应用数学方面都有杰出的贡献,但他对人类更杰出的贡献是在电子计算机领域。冯·诺伊曼对世界上第一台电子计算机ENIAC(电子数字积分计算机)的设计提出过建议,1945年3月在与他人共同讨论的基础上起草了EDVAC(电子离散变量自动计算机)设计报告初稿,在这个新方案中,他提出了两个至关重要的设想:存储程序和采用二进制系统。程序预存储使运算自动化,而二进制系统大大简化了电路,使运算速度加快,这对后来计算机的设计有决定性的影响。特别是确定计算机的结构,采用存储程序以及二进制编码等,至今仍为电子计算机设计者所遵循。冯·诺伊曼关于电子计算机的设想奠定了现代电子计算机的计算模式基础,因而被称为现代电子计算机之父。

冯·诺伊曼计算机体系结构的要点有:

- 计算机中的信息(程序和数据)以二进制方式表示。
- 程序预存储,机器自动执行。
- 计算机由运算器、控制器、存储器、输入设备和输出设备五大部分组成。

1. 中央处理器(CPU)

中央处理单元(Central Processing Unit, CPU)也称为“处理器”,是计算机的核心部件,其重要性好比大脑对于人一样。其主要功能是解释并执行计算机指令,完成数据处理和对计算机其他各部分进行控制。

CPU主要由运算器、控制器、寄存器组和内部总线等构成。其中运算器主要完成诸如加、减、乘、除等算术运算和左右移位、与、或、非等逻辑运算;控制器主要完成机器指令的解析和执行,控制运算器进行相应的运算,控制数据在计算机各组成部分之间传送,控制计算机各组成部分有条不紊地协调工作等;寄存器组是处理器内部的一组存储单元,一般又可分为两类,即数据寄存器(主要用于暂存数据处理过程中的中间结果)和专用寄存器(用于标志系统的状态和设置系统的工作方式等)。CPU内部的这些功能部件通过内部总线互联并构成一个有机的整体。

CPU依靠执行指令来完成计算和对系统进行控制。每款CPU在设计之初就规定了一套与其硬件电路相配合的机器指令。人们将一款CPU所支持的机器指令全集称为这款CPU的“指令系统”,这些机器指令以二进制数据编码方式存在,CPU能够直接译码并执行这些以二进制方式存在的机器指令。人们使用这些指令编程控制计算机的运行,人们编写的用于控制计算机工作的指令序列称为计算机程序,计算机执行程序完成数据的处理。

CPU执行指令的过程大致可分为以下三个阶段:

- 取指(Fetch):将待操作的指令码从存储器取到处理器。
- 译码(Decode):对指令码进行分析译码。
- 执行(Execute):产生指令码对应的操作信号。

在CPU中有一个特别用途的寄存器即PC(Program Counter,程序计数器),用于指示待取指的内存单元。首先,CPU自动地将PC所指向存储器单元里存放的机器指令码取到CPU中,同时移动PC指向下一条待取指指令码的内存单元(这个过程称为取指);接着,控制器

的指令译码电路即对这条指令进行分析(译码);之后,生成这条指令功能对应的控制时序,完成本条指令的功能操作(执行);处理器再对下一条指令递归上述的取指、译码、执行各步骤,CPU 依次执行组成程序的各条指令,完成程序描述的功能,如图 1-1 所示。

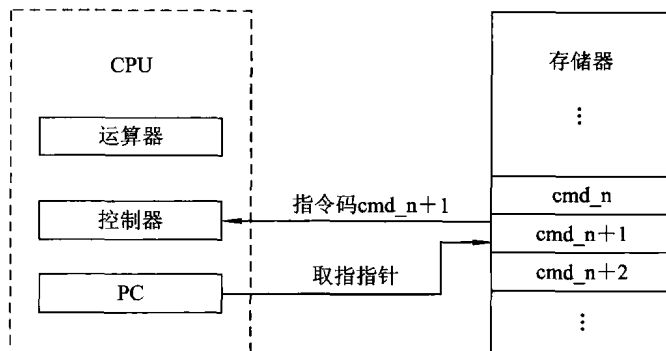


图 1-1 处理器执行程序示意图

有两个基本参数用于描述 CPU 的性能:主频和字长。

- CPU 主频:也叫时钟频率,单位是 MHz(或 GHz),用来表示处理器的工作频率。CPU 主频很大程度上决定了 CPU 的运算、处理数据的速度。需要对初学者说明的是,处理器的工作频率并不能完全决定计算机系统的工作性能,计算机系统的性能还与字长、体系结构、设计方法、运行环境等因素相关。

- CPU 字长:即运算器的位宽,单位是比特(bit),用于表示 CPU 一次运算可处理数据的数据位宽度。

2. 存储器(Memory)

存储器(Memory)是计算机系统中用来存储程序和数据的记忆部件。有了存储器,计算机才有记忆功能,才能正常工作。计算机中的全部信息,包括输入的原始数据、计算机程序、中间运行结果和最终运行结果都保存在存储器中。在处理器控制下,可以向地址指定的存储器单元存入数据(也称为写存储器)或从地址指定存储器单元取出数据(也称为读存储器)。按用途存储器可分为主存储器(内存储器,简称内存)和辅助存储器(外存储器,简称外存)。外存通常是磁性介质或光介质等存储器,能长期保存信息,存储容量大、速度慢,单位容量成本低,掉电后存储的信息不丢失;处理器通过专用的接口电路访问外存数据。内存是指计算机主板上的存储器部件,用来存放当前正在使用的数据和程序,处理器通过系统总线访问内存数据;内存通常用高速半导体器件实现,速度快、单位容量成本高。目前主流的内存多使用随机存储器(RAM)、闪存(Flash)或只读存储器(ROM)实现,RAM 存储器掉电后存储的信息会丢失。

计算机的存储器通常以字节(Byte)为单位进行组织和编址,处理器通过地址指定访问存储器各单元。1 字节由 8 个 1/0 存储位(bit)组成。实际应用中,用于描述存储器容量常用的单位有 B(Byte)、KB、MB、GB 和 TB 等。其中,1 KB = 2^{10} B,1 MB = 2^{10} KB,1 GB = 2^{10} MB,1 TB = 2^{10} GB。

3. 输入/输出接口与设备(IO Device)

计算机通过执行程序完成信息的处理,这就需要有通道能将程序和待处理的数据送入

到计算机中, 计算机处理后的结果数据也能够通过某种通道被送出来。这些完成信息进出计算机的专用设备称为输入/输出设备(简称为外设)。常见的外设有键盘、鼠标、显示器、打印机、光驱、硬盘等。

外设通常是一些机电装置, 它们的信息表示形式种类繁多, 而计算机所能接受和处理的信息表示形式只能是高低电平; 再者, 这些机电设备的速度与处理器的速度差别很大, 故而外设在与计算机连接时, 一些信号需要进行转换和控制。输入/输出接口是用于外设与计算机之间桥接和匹配的专用部件, 通常以集成电路或板卡的形式存在。

在计算机原理中, 我们所说的外设主要是指这些输入/输出接口(简称为接口外设), 它不特指某一具体的外设物理装置, 更具有逻辑原理普遍性。常见的外设接口有并口、串口(RS232、USB、I²C 等)、定时器等。

1.1.4 计算机系统

计算机的物质组成部分称为计算机硬件。计算机硬件是一套复杂的电子、机械组合体, 仅有硬件的计算机称为裸机。裸机仅提供了功能强大的躯体, 还远不足以完成复杂的信息(数据)处理, 还需要人们通过编程的方式告诉它所要处理的数据、数据的处理方法以及计算机中所有资源(CPU、存储器、外设、任务、文件)的管理方法等, 计算机才能按照人们编程预约的工作策略、步骤进行工作。我们将编程方式描述的数据和数据处理方法称为计算机软件。将计算机软件和硬件合称为计算机系统。赋予了软件的计算机才能胜任智能化的工作。

计算机软件按照其功能层次分为系统软件和应用软件。系统软件是指那些通用的、基础的计算机资源管理软件和计算机基础功能软件, 主要是操作系统软件, 如 Windows、Linux、μC/OS 等操作系统, 用于管理、协调计算机中的所有资源, 包括硬件资源, 也包括数据、文件, 还包括用于完成数据处理的任务等。通过操作系统的封装, 使人与计算机的交互更为友好, 各部分的工作更为协调, 人们使用计算机更为方便, 系统工作效率更高。应用软件则是指那些为解决某一特定问题而编制的用户专用程序和数据, 如图书馆管理软件、手机监控程序、数码相机监控程序等。

计算机系统的组成参见图 1-2。

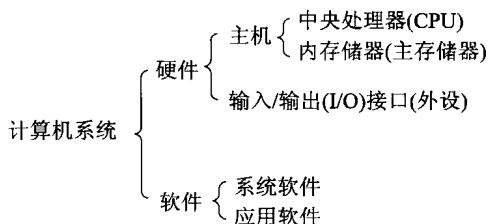


图 1-2 计算机系统的组成

1.1.5 计算机的工作过程

电子计算机自诞生到现在, 主流计算机的结构仍沿用冯·诺伊曼计算机体系结构。尤其是在微型计算机中, 计算机硬件各个部分都是通过总线互联的, 这就是所谓的总线结构

型计算机。所谓的总线(Bus)就是连接计算机硬件各部件,用于计算机硬件各部件之间信息传输的公共通道。物理上它就是一组公共信号线,可能是一组明线,也可能是一组 PCB 板上覆铜线。按照其传送信号的用途属性,总线可细分为地址总线(Address Bus, AB)、数据总线(Data Bus, DB)和控制总线(Control Bus, CB)三类。

- 地址总线:专用于在 CPU、存储器和 I/O 端口间传送地址信息的信号线。此类信号线传送的信息总是从 CPU 到存储器或 I/O 端口,它是单向信号线。

- 数据总线:专用于在 CPU、存储器和 I/O 端口间传送数据信息的信号线。此类信号线传送的信息可以是 CPU 到存储器或 I/O 端口(“写”操作),也可能是从存储器或 I/O 端口到 CPU(“读”操作),它是双向信号线。

- 控制总线:专用于 CPU 与其他部件之间传送控制信息和状态信息的信号线。此类信号线的构成比较复杂,传送的控制、状态信息可以是 CPU 到其他部件,也可能是从其他部件到 CPU。此类总线中的某些具体的线是单向的(或从 CPU 到其他部件,或反之),但作为总线来说,它是双向信号线。

从硬件上看,微型计算机就是一个以 CPU 为核心部件,以三总线为纽带将计算机各部件互联起来的能自动完成数据处理的数字系统,如图 1-3 所示。

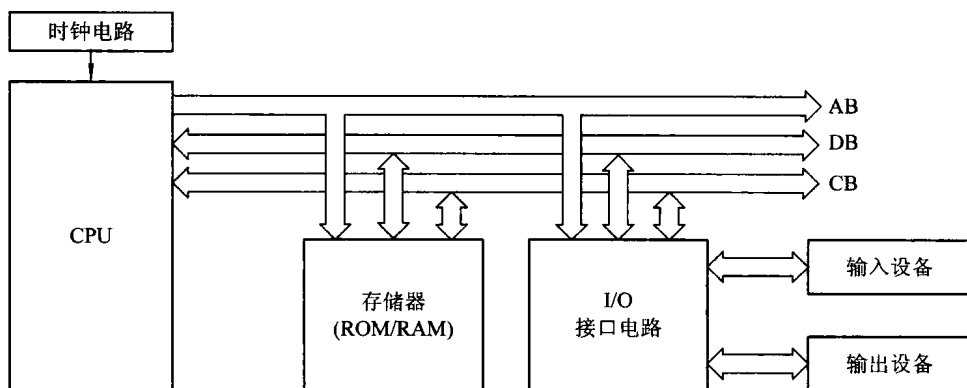


图 1-3 微型计算机的硬件系统结构

计算机的工作过程就是处理器执行预存于存储器中的程序,在程序的控制下对数据信息进行加工处理的过程。人们预先要把指挥计算机如何进行操作的指令序列和原始数据(合称为程序)通过输入设备输送到计算机的内存储器中。每一条指令中明确规定了计算机从哪个地址单元取数,进行什么运算,然后送到哪个地址单元中等操作。计算机在运行时,先从内存中取出第一条指令,通过控制器的译码,按指令的要求,从存储器中取出数据进行指定的运算处理,然后再按地址把结果送到内存中。接下来,再取出第二条指令,在控制器的指挥下完成规定操作,并依次进行下去。

1.2 数制和计算机中的数值表示与处理

计算机中的所有信息都是以二进制的形式表示的,数值量是计算机中要表示的最基础的信息。数值具有量的概念。在计算机记录和运算的过程中,数值在计算机中的表示方法

称为数据。数据不但能表示数值，还能表示其他的信息。实际上，计算机中的所有信息(指令、数值、文本、音像等)都是以二进制数据的形式存在的，所以我们将它们统称为数据。对数值的计算机表示和处理是计算机的基本功能，在此我们仅介绍计算机基础数据——整数数值的表示方法和处理规则；复杂信息数据(如浮点数、音像数据等)在计算机中的表示和处理是编码学等专业课程的内容，事实上它们也是以二进制数据为基础的，我们不在此讨论。

计算机中之所以采用二进制数制，是因为用数字电路的高/低电平易于实现二进制数的表示及运算，并且二进制数通过编码可以表示任何信息。

日常生活和工作中人们习惯使用十进制数制进行数值表示与运算，而计算机又是以二进制为基础的。为此，我们在此仅讨论如下几种数制的数值表示和运算。

1.2.1 无符号数的十进制、十六进制、二进制表示法与运算

1. 十进制数表示法和运算规则

日常生活中，人们习惯使用十进制计数体制。它的基本规则是：

(1) 用 10 个基本符号集{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}中的元素(称为基数)排列表示数据，并依此表示数值。

(2) 加法运算时“逢十进一”，减法运算时“借一当十”。

任何一个十进制数 N 都可以表示为：

$$N = \sum_{i=-m}^{n-1} D_i \times 10^i \quad (1.1)$$

其中， D_i 为第 i 位上的基数， 10^i 为“权位”。例如：

$$123.45 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}$$

十进制数的加、减运算不再赘述。十进制数是默认的计数体制，十进制数据表示时可以不作特别的标注，也可以用后缀“D”(或“d”)显示标注。如数据 123.45，默认为十进制数据，也可以显示表示为 123.45D 或 123.45d 等。

2. 十六进制数表示法和运算规则

十六进制数的基本规则是：

(1) 用 16 个基本符号集{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}中的元素(称为基数)排列表示数据，并依此表示数值。

(2) 加法运算时“逢十六进一”，减法运算时“借一当十六”。

为了与十进制数书写有区别，在十六进制数据的尾部加上 H(或 h)，以标注这是一个十六进制数。如 123.45H 或 123.45h，表示它是一个十六进制的数据。(在本书其他部分用前缀“0x”表示十六进制数据，如 123.45H 在本书中表示为 0x123.45，以符合 ARM 技术资料的表示习惯。)

任何一个十六进制数 N 都可以表示为：

$$N = \sum_{i=-m}^{n-1} D_i \times 16^i \quad (1.2)$$

其中, D_i 为第 i 位上的基数, 16^i 为“权位”。例如:

$$\begin{aligned} 1A3.D8H &= 1 \times 16^2 + A \times 16^1 + 3 \times 16^0 + D \times 16^{-1} + 8 \times 16^{-2} \\ &= 1 \times 16^2 + 10 \times 16^1 + 3 \times 16^0 + 13 \times 16^{-1} + 8 \times 16^{-2} \\ &= 419.84375 \end{aligned}$$

即 1A3.D8H 就是十进制数值 419.84375 的十六进制表示。

例 1.1 十六进制数加、减运算: ① 1DF.Ah + 4C.3h; ② 1DF.Ah - 4C.3h。

解

①

$$\begin{array}{r} 1DF.Ah \\ + 4C.3h \\ \hline 22B.Dh \end{array}$$

②

$$\begin{array}{r} 1DF.Ah \\ - 4C.3h \\ \hline 193.7h \end{array}$$

3. 二进制数表示法和运算规则

二进制数的基本规则是:

(1) 用两个基本符号集 {0, 1} 中的元素(称为基数)排列表示数据, 并依此表示数值。

(2) 加法运算时“逢二进一”, 减法运算时“借一当二”。

使用二进制数表示数据时, 在二进制数据的尾部加上 B(或 b), 以标注这是一个二进制数。如 101.01B 或 101.01b, 表示它是一个二进制的数。

任何一个二进制数 N 都可以表示为:

$$N = \sum_{i=-m}^{n-1} D_i \times 2^i \quad (1.3)$$

其中, D_i 为第 i 位上的基数, 2^i 为“权位”。例如:

$$101.01B = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 5.25$$

即 101.01B 就是十进制数据 5.25 的二进制表示。

例 1.2 二进制数加、减运算: ① 1101.101b + 100.1b; ② 1011.1b - 101.01b。

解

①

$$\begin{array}{r} 1101.101 \\ + 100.1 \\ \hline 10010.001 \end{array}$$

②

$$\begin{array}{r} 1011.1 \\ - 101.01 \\ \hline 110.01 \end{array}$$

4. 数制之间的转换

同一个数值在不同的数制中有不同的数据表示形式。在此要注意: 仅仅是表示的形式不同, 而它们的数值含义是相同的。日常生活中, 人们习惯使用十进制计数体制, 而计算机内使用的是二进制数, 同一数值在不同数制中有不同的表示方式, 这就需要对二进制数与十进制数进行转换。二进制数的权基数比较小, 在表示一个较大的数值时, 数据码串比较长, 人们识别起来不方便, 并且转换成十进制数时项数比较多, 转换运算量也较大。二进制数与十六进制数有直接的按位对应转换关系, 转换起来极为方便。十六进制数权基数比较大, 表示一个较大的数值时, 数据码串较短, 人对其较易识别, 与十进制数转换时运算也比较方便。十六进制数兼顾了二进制数和十进制数的特点, 在计算机编程中是一种常用的数据表示方式, 也常作为一个理想的中间工具, 用于二进制数与十进制数的数据转换。

1) 二进制数与十六进制数之间的转换

在数值上 $2^4=16$, 故在二进制数与十六进制数之间进行转换时存在按位的对应关系, 转换很方便。二进制数与十六进制数的对应关系见表 1-1。

表 1-1 二进制数与十六进制数的对应关系

二进制数	十六进制数	二进制数	十六进制数
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

十六进制数到二进制数的转换规则: 以小数点为参照, 分别转换整数部分和小数部分, 每 1 位十六进制数直接转换为 4 位的二进制数即可。

例 1.3 将 1FA.Ch 转换为二进制表示。

解 $1FA.Ch = 0001\ 1111\ 1010.1100b = 1\ 1111\ 1010.11b$

$\underbrace{\hspace{1cm}} \quad \underbrace{\hspace{1cm}} \quad \underbrace{\hspace{1cm}} \quad \underbrace{\hspace{1cm}}$
 1 F A . C

二进制数到十六进制数的转换规则: 以小数点为参照, 分别转换整数部分和小数部分, 整数部分向左(小数部分向右)每 4 位二进制数划分为一个转换单元(不足部分补 0), 将 4 位二进制数直接转换为 1 位十六进制数即可。

例 1.4 将 10 1011 0101.011b 转换为十六进制表示。

解 $10\ 1011\ 0101.011b = 0010\ 1011\ 0101.0110b = 2B5.6h$

$\underbrace{\hspace{1cm}} \quad \underbrace{\hspace{1cm}} \quad \underbrace{\hspace{1cm}} \quad \underbrace{\hspace{1cm}}$
 2 B 5 . 6

2) 二进制数、十六进制数到十进制数的转换

此类数据转换只是计算的问题, 分别按照公式(1.2)、(1.3)将十六进制数或二进制数展开, 再按照十进制数运算法则计算, 即可直接计算出相应的十进制数。

例 1.5 将 1FA.Ch 转换为十进制表示。

解 $1FA.Ch = 1 \times 16^2 + 15 \times 16^1 + 10 \times 16^0 + 12 \times 16^{-1} = 506.75$

例 1.6 将 10 1011 0101.011b 转换为十进制表示。

解 $10\ 1011\ 0101.011b = 1 \times 2^9 + 0 \times 2^8 + 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3$
 $+ 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$
 $= 693.375$

备注: 从此例可以看到, 当一个比较大的二进制数转换为十进制数时, 运算起来比较麻烦, 如果先将二进制数转换为十六进制数, 再从十六进制数转换为十进制数会方便些。
即

$$10\ 1011\ 0101.011b = 2B5.6h = 2 \times 16^2 + 11 \times 16 + 5 \times 16^0 + 6 \times 16^{-1} = 693.375$$